A Research Proposal

on

An Automated Trust-based Security Framework for Maximizing Authorization control in Web Services

Submitted to

LOVELY PROFESSIONAL UNIVERSITY

in partial fulfillment of the requirements for the award of degree of

DOCTOR OF PHILOSOPHY (Ph.D.) IN Computer Applications

Submitted by:

Rohit Dhand

Reg. No. :40900058

Supervised by:

Dr. G. Geetha

Associate Prof.,

Deptt. Of CSE/IT, LIECA

LPU.

LOVELY FACULTY OF TECHNOLOGY AND SCIENCES

LOVELY PROFESSIONAL UNIVERSITY

PUNJAB

1. INTRODUCTION

Prevailing trends in modern era of computing has shifted everything to web. Web which started as a very small concept of file sharing on the network with HTML developed by Tim Berner Lee, has now changed the way we interact with each other. Over the last decade computing with respect to internet and web has surpassed all kinds of growth indices. There are lots of examples to prove the above fact like: Facebook, Twitter, Gmail, Amazon, Microsoft Office WebApps etc. The innovation in the Web standards and growth of internet usage has led to the development of various technologies that are replacing the traditional personal computing technologies. The Web also opened various opportunities for replacement of a PC with very innovative mobile devices like: mobile phones, tablets, PDA's, iPads, iPhones **[1]** etc. to access the web and services.

Another major domain area where there has been a tremendous change is the availability of services over the internet. Traditional services like Email, Search Engines are being extended to Social Networking, Cloud Computing, and SaaS etc. Major technology giants like Google, Microsoft, Amazon, Apple, IBM and others are investing heavily in the technologies that are going to redefine and keep their supremacy intact in the world of web. The widespread emergence of the Web Technologies and success of the companies like Google, Facebook etc. has accelerated the adoption of a platform for electronic data distribution and the advent of structured information. While the introduction of Extensible Markup Language (XML) as a structured format was a major enabling factor, the promise offered by SOAP based web services triggered the discovery of architectural patterns that are now known as **Service Oriented Architecture (SOA). [2]**

Service Oriented Architecture (SOA) is one domain that is emerging very rapidly and changing the way we write web applications. Recent years have seen a tremendous growth of the service-oriented applications **[3, 4]**. More and more companies are investing heavily in Service Oriented Architecture. The principal benefit which the companies derive by implementing Service Oriented Architecture is, to redefine the existing infrastructure and reforming the legacy applications as consumable services. This helps them to reduce the cost of maintaining the system and support the end users with legacy applications over the network. Service Oriented Architecture contains the functionality of interoperable services also known as **Web Services**. These services are platform independent and they can be consumed over a network. For the end user, it's

basically a browser or an application that calls the web service keeping abstraction phenomena working at the backend.

Several new trends in the computer industry rely upon SOA as the enabling foundation. These include the automation of Business Process Management (BPM), design patterns of **Web 2.0.** Web 2.0 can be generally characterized as a common set of architecture and design patterns, which can be implemented in multiple contexts. The list of common patterns includes the Mashup [5], Collaboration-Participation, Software as a Service (SaaS) and Rich User Experience (also known as Rich Internet Application) patterns among others. Most Web 2.0 architecture patterns rely on Service Oriented Architecture in order to function.

Service Oriented Architecture has the potential to make the concept of reusability real. There have been lot of talks over the years about reusability and its implementation, but nothing concrete has been done to convert this concept into a reality. Typically an IT manager in any business house does not know how many duplicate processes are being run in an ERP system. So by using Service Oriented Architecture the business houses can identify the overlapping processes in the system. Figure 1.1 [3] shows an example of an information system scenario that could benefit from a migration to SOA. It shows the separate business processes using the same functionality with respect to login. The company is basically paying three times to implement the same functionality.

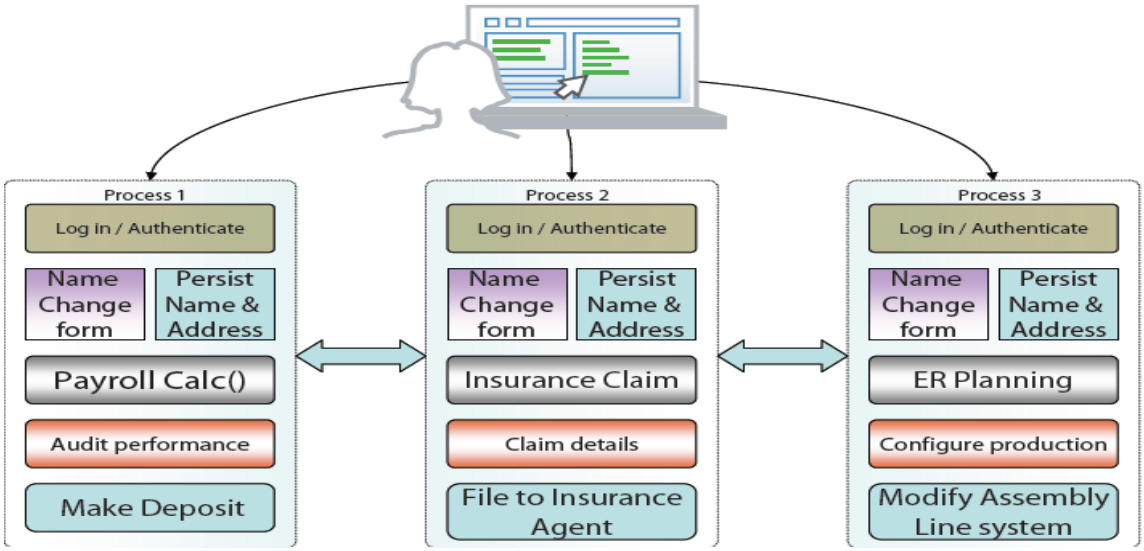

Figure 1.1 (IT Infrastructure without SOA)

In a more efficient scenario, common tasks would be shared across all three processes. This can be implemented by decoupling the functionality from each process or application and building a standalone authentication and user management application that can be accessed as a service. This would be a simple example of SOA. The resultant IT infrastructure would resemble as of figure 1.2

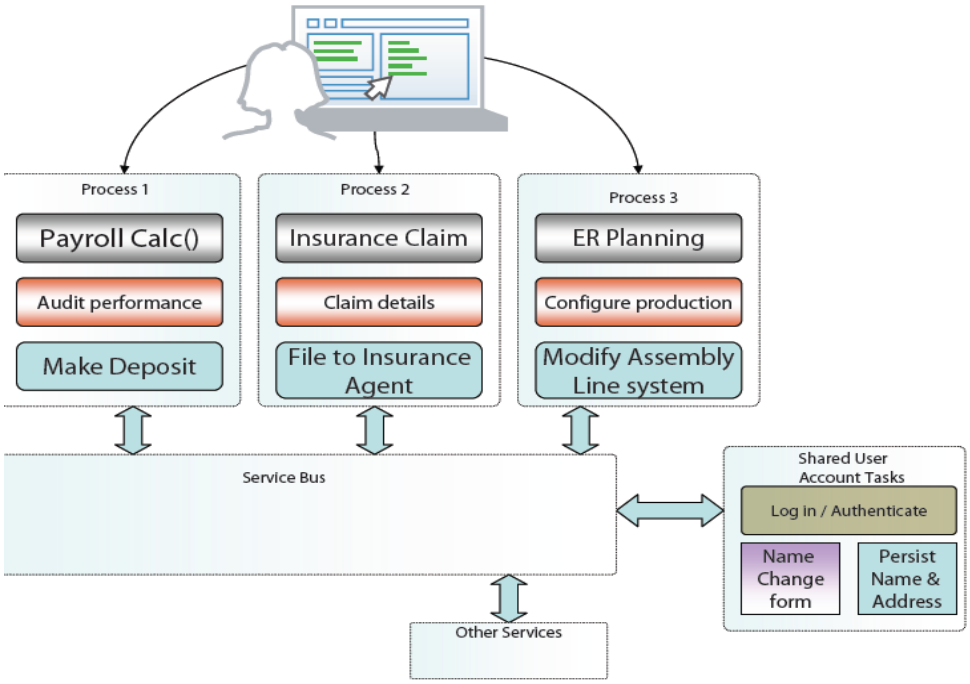

Figure 1.2 (IT Infrastructure with SOA implementation)

In the preceding example of SOA, the complications were relatively minor as the entire infrastructure existed within one domain. In reality, enterprise SOA is much more difficult because services may be deployed across multiple domains of ownership. SOA defines how to integrate widely disparate applications for a Web-based environment and uses multiple implementation platforms. Rather than defining an API, SOA defines the interface in terms of protocols and functionality. That's why they are an extension to the traditional distributed computing model and a strong foundation layer for cloud computing [6].

By implementing SOA, the companies can run more effectively. Companies with large legacy back-end systems, like Google, Amazon have used this approach to modernize existing infrastructure. This has also helped the companies in Web 2.0 so that they can grant access to data that was (possibly) expensive to collect, allowing 3rd party

developers to create new applications using this data. These services are commonly referred to as web services, as they are usually available via the web.

1.1 Background

Service Oriented Architecture is implemented with the help of Web Services. In fact Web Service is a peace of code/logic that runs on the Web Server and handles the client request from any part of the world. **Web services are trusted [7]** software components that communicate using pervasive, standards-based Web technologies including HTTP and XML-based messaging. Web services are designed to be accessed by other applications and vary in complexity from simple operations, such as checking a banking account balance online, to complex processes running CRM (customer relationship management).

Web services are hardware, programming language, and operating system independent. This means that applications written in different programming languages and running on different platforms can seamlessly exchange data over intranets or the Internet using Web services.

Web services are powered by the following core components:

a. XML
b. WSDL
c. SOAP
d. UDDI

XML (eXtensible Markup Language)

XML [8, 9, 10] is a W3C (World Wide Web Consortium) specification that defines a meta-language for describing data. In XML applications, data is described by surrounding it with customizable, text-based tags that give information about the data itself as well as its hierarchical structure. Because of the flexibility in the tags that XML provide, it has helped XML achieve widespread acceptability and adoption as the standard for exchanging information between heterogeneous systems.

WSDL (Web Services Description Language)

WSDL is an XML-based format for describing Web services. Clients wishing to access a Web service can read and interpret its WSDL to learn the details about the Web Service

and the operations it provides. WSDL thus acts as the initial interface/contract through which the clients can interact with the web service by following a standard process.

SOAP (Simple Object Access Protocol)

SOAP **[11]** is an XML-based protocol recommended by W3C for exchanging data over HTTP. It provides a simple, standards-based method for sending XML messages between applications. SOAP [12] is commonly used in Web services. Because HTTP is supported by all Web servers and browsers, SOAP messages [13] can be sent between applications regardless of their platform or programming language. SOAP message includes the message, header, envelope, body, fault etc.

UDDI (Universal Description Discovery and Integration)

UDDI **[8]** is a standard sponsored by OASIS **[14]** (Organization for the Advancement of Structured Information Standards). UDDI is often described as the yellow pages for the web services. It provides the businesses a uniform way of listing their services and discovering services using WSDL

In brief, the web services have got the following characteristics/features:

- ✓ Web Services are purely logic/code and does not contain any user interface.
- ✓ Web Services are hosted on the web servers.
- ✓ Web services can be invoked by any developer through SOAP i.e. Simple Object Access Protocol.
- ✓ Web Services exchange the data on the basis of fixed XML Schema.
- ✓ Web Services use the WSDL (Web Services Description Language) for publishing and accessing them through the meta-data provided by WSDL.
- ✓ Web Services are purely platform independent.
- ✓ Web Services contains loosely coupled public methods that perform different actions according to the business verticals.

So, Web Services are making the lives of the developers more and more convenient and providing a better platform for application development. As the penetration of the Web Services is increasing day by day, serious security problems **[15]** are also threatening the Service Oriented Architecture [**16**, **17**]. Primarily the threats can be classified into the following:

a. Outside Domain: Like sniffing, hacking, session hijacking etc. **[18,19,20,21]**
b. Inside Domain: Within a business domain like Unauthorized Access **[22]** to the web services

Both categories of above mentioned threats are serious in nature. With lots of Web Services being written and hosted on a variety of platforms, the problem of creating a reasonable **combination of accessibility and access restrictions arises among the administrators**. **The access to the web services and its services functions must be controlled and authorized [23] because of the following:**

a. **More and more web services are written and they can be misused by an unauthorized client/user.**
b. **Web Services normally access the databases at the back end. This results in ensuring that the web services unauthorized access should be strictly blocked.**
c. **Employing an IT administrator to do the task of administration of Access Control looks, out dated and cannot replicate in the large setup where number of users and resources to be configured are in a big volume.**

Although the authentication and authorization of the Web services is managed by some kind of human intervention i.e. a System administrator, but in a longer run the exorbitant amount of Web services being administered through a manual system will make the job more tedious and complex affair. Setting up of the access rights and role based access through a system administrator will look out to be a mammoth job. So there arises a need for the development of a system that can automate the system of authorization control to an extent so that the system itself takes care of the majority of the authorization jobs. But this does not mean that the human intervention will go away. It will always be needed to provide another layer of flexibility in order to customize the access restrictions, if necessary. In fact there can be a semi-intelligent framework [24] system that works on certain parameters to dynamically manage the access control and restrictive access [22] in web services. There must be some parameters to define the system. One of the base parameters is **"Trust" [25, 26, 27]**

Meaning of Trust

Trust **[28]**, is a very important aspect of human life. We use it every day in one way or another. For e.g. we trust our car for travelling to our office every day, we trust our bank to give us the money we claim in the future. Trust is central to all transactions, where our

own actions are dependent on the actions of others. When we say we trust someone or that someone is trustworthy, we implicitly mean that the probability that he will perform an action that is beneficial or at least not detrimental to us is high enough for us to consider engaging in some form of cooperation with him. Correspondingly when we say that someone is untrustworthy, we imply that the probability is low enough for us to refrain from doing so. In fact the trust can be represented with a range from 0 to 1, where 1 represents complete or blind trust and 0 is complete distrust **[28]**. With above discussion in mind, trust can have the following properties:

1. *Trust is relativized to some transaction* **[29]**. A may trust B to drive a car but not to sell the card.
2. *Trust is a measurable belief.* A may trust B more than A trusts C for the same business.

Establishing Trust

Trust relationships are usually based on identity. Information about an agent's behavior that was gained earlier can only be applied in later transactions, if the agent is recognized as the very same. Thus a working authentication system is essential to establish a trusted environment. This respect is the most limiting part in the adaption of trust in computer systems. A user who must obtain a security token, get a private key before he can use a specific service. In this identity-based trust, a trust relationship will be of a strong or weak value.

My research problem's approach will try to mimic the decisions taken by humans when it comes to judging whether an action of an opposing party is beneficial or not. The objective is to create a collection of Web Services where security is assured both by *standard safety techniques for transport and a trust based approach for access control.*

Trust related projects

A detailed analysis and look at the trust related projects is not easy job because there are no existing standards on this issue at the moment. Efforts have been made by major companies to standardize something based on Trust. The major outcome of all efforts is worth mentioning and known as **"WS-TRUST".**

WS-TRUST provides a mechanism to obtain a Security Token and establish a trusted relationship between two parties. The current working of the WS-Trust purely works in the ***static concept i.e. Complete Trust/Blind Trust or Complete Distrust***. For e.g. a requestor sends a request, and if the policy permits, the recipients request is entertained after receiving a security token. Otherwise the request of the requestor is declined. The

standard also defines a hierarchical structure to fetch the token for a particular user from a *trusted root/source.* WS-Trust works in a very special structured approach and is an important issue in the problem being presented in the synopsis. However, WS-Trust will not be able to perform very effectively in a cluttered environment. So that's why there is another standard which is worth mentioning i.e. **"WS Federation Language".**

The WS-Federation Language is based on the trust where multiple federations of web services are going to participate. The mechanism will allow the trust of identities, attributes and authentication between federations of web services. In this every user will have a **single sign-on**. Once is he/she is authenticated, he can use the other portals without authentication because the portals are linked through with the help of their identity providers. So the data is passed/exchanged from one portal to another.

An example for a project dealing with **role assignment based on a user's behavior is the TERM Server Architecture**, **done at Purdue University's Center for Education and Research in Information Assurance and Security (CERIAS**, [30]). This project implements access control based on **direct and recommended trust**. Furthermore it tries to establish a **standard for trust in Computer Systems and deal with the arising problems for which there is no satisfying solution at the moment.**

2. REVIEW OF LITERATURE

Recent efforts by several researchers have led to a plethora of new and innovative techniques for security in the Service Oriented Architecture. The exponentially growing field of Web Services led to more and more researchers doing research in the field of Security in the Web Services. There have been a lot of efforts on the development of standards within the Industry and academia on the security techniques to deal with the problem of authentication and authorization in Web Services.

In fact the security initiatives with respect to the web services started in the early **2003**

Research initiatives during the year 2003-2007

The first kind of security in the mobile computing/distributed computing was highlighted by **Hua Wang [31].** There was a proposal of a framework that was based on tickets. The paper proposed a secure and flexible access control scheme and protocol for role based access control. The architecture was based on a Trusted Credential Centre, A Trusted

Authentication and Registration Center that will issue tickets based on the authentication. Also the tickets govern the services or access control which the end user/client can access.

David Geer **[11]** highlighted the importance of **SAML (Secured Assertion Markup Language)** developed by Organization for the Advancement of Structured Information and Standards (OASIS). Authorization control which is based on role based model was highlighted by **Alfred C. Weaver [1]**. A suggestion towards a context conditions was made by him. According to the rule any request which comes for certain access, must go through the constraint checking and evaluated on those conditions. The context implementation of a context type is a defined as program or function that can evaluate the value of the context type. This may be implemented for access control, policies, boolean results etc. Another major bold initiative was taken by **Microsoft Research Division**. They in fact developed a tool called **"TulaFale"** for the security of the web services. The tool provided a way out to encrypt the messages and headers of the SOAP packets.

"SecureZone" [32] was another major framework for WS-Secure and WS-Trust based security. The above framework was also accepted by the **Apache Foundation,** to be implemented in the Apache Web Server. In this a Trust Engine was suggested that will receive the requests and evaluate them against the trust level and then granting of the various resources is allowed.

The first effort that was based on trust came into existence under the name of Trustworthy computing, was suggested by **Jia Zhang, Liang-Jie Zhang, Jen Yao Chung [33]**. They suggested a complete framework that could be based on any organization, users, roles, rules and processes. This framework was the first step in the security of the web services based on trust. Working on the same pattern, Lalana Kagal and Tim Finin **[22]** suggested the importance of privacy and authorization control. They suggested the methods of policies that could be used for authorization control.

A real effort was made by Pakpoom Prechapanich [8] where the Web Services Security Framework for the Ministry of Information and Communication Technology [Thailand] was developed. It was coded as **BS7799** a university acceptable standard and also used the **COBIT** framework. The result of this framework was quite encouraging. The framework was based on the following aspects of the domain:

a. Security Policy
b. Organizational Security
c. Asset Classification and Control
d. Personnel Security
e. Access Control.
f. Business Continuity Management
g. Compliance.

The above framework was a good achievement in the method of securing web services. But it lacked the trust factor in building of the framework. **Token Based Dynamic Trust Establishment for Web Services suggested by Alfred. C. Weaver [34]** was another major step forward towards the authentication and authorization with web services. Here the idea was to develop a technique in which the privacy of the client is observed by not compromising on the security of the web service. In this method class client will dynamically discover the policy document and reveal its attributes as required by the functions of the web service. Thus ensuring the client is not compromising on its privacy features. Lots of other methods based on token registration, encryption of SOAP headers and messages etc. [35, 36] were suggested with some enhancements to the already existing methods.

With the various enhancements in the web services security, a new method of security in web services with help of *indirect trust* was suggested by **Zhengping Wu and Alfred C. Weaver [37].** The indirect trust will help in other participating agents in the web services framework to get recommendations from nodes/agents that have direct trust with agents. They developed a tool by using .NET that would help in the implementation of the same. In fact centralized trust models a common trusted intermediary called "Trust Authority". Trust Authority is used for establishing the trust between two participating users/agents. The trust authority would be a central authority that would check the validity of the requests from multiple users and then giving access rights to access them.

Another attempt in the direction of security of public web services framework was made by J.Thelin [21] who suggested the concept of **CapeConnect** authentication service. The clients authenticate themselves from this server and obtain a session ticket. The session ticket is used to control the credentials and access control till the user logs out.

Loubna Mekouar and Youssef Iraqi **[38]** suggested **TrustWS**- A trust management system for web services. They proposed a system of Trust Manager that will allow the users to select the appropriate web services based on the feedback from past transactions. It was the first major effort on building a system that can really work on feedback / past record.

Measuring and testing trustworthiness **[39]** of the web service also came into existence. There were suggestions on implementing algorithms that can measure the trustworthiness of the web service security. A number of activities on building of efficient access of web services **[40, 41]** and trust negotiations in identity management came up. The bootstrapping concept in trust establishment which provided the initial index values was also suggested by various frameworks. These were used in multiple business domains like E-Commerce [42], M-Commerce **[43]** etc.

Research initiatives during the year 2008

Year 2008 saw further extensive researches in the field of Trust based security in web services. Jinpeng Wei [2] suggested a mechanism in which there was a proposal for the division of Web Services Platform into two domains: (a). A small Trusted T-WSP to handle the sensitive data. (b). Large scale functionality for the use of WSP. Usage of the sensitive information will be governed by the T-WSP. In fact this was suggested to tackle the problem for the sensitive information like, username/password, credit cards, bank account information etc. A tool was developed that could handle the request to ensure the authorization of the sensitive data based on upon the trust level.

A similar kind of tool was developed in **Java by Walter Binder [44]** which basically took certain parameters so as to support user defined selection and ranking functions to access various resources in the web services directory. Trust algorithms were a now a hot topic and lot of researchers started proposing trusted algorithms. Trusted algorithms were also adopted in the development of **"Kerberos" [45]** an authentication system developed by **MIT**. The same system has been implemented in a very popular operating system known as Microsoft Windows Server System developed by Microsoft.

A Security based on the token mechanism was suggested by Nian Liu [46]. Although it was meant for the Wind Power Plants, but the same can be replicated to any system and

business domain. It was based on the PKI system. Enhancements in trust algorithms were also suggested by R. Yahalom [28] and B. Klein for evaluating the requests of clients. The algorithms work on the classification of the trust types and level that two agents in the system collaborate on.

Research initiatives during the year 2009

The research was further contributed by Social Network based trust for agent based services [47, **48**, 49]. The idea was to derive a network (social network) that can keep track of the trust between the participating agents. This can help in authentication and verification. Various T-Index approaches were also developed and suggested so that it can resolve the problems of administration of web services. **Trustworthy web services can greatly enhance the productivity of administering the web services. Concept of Third Party Trust, Social Trust, and matchmaking based trust were coming into various frameworks.** These frameworks were able to improve the efficiency with regards to the management of the service oriented systems. Even some researchers suggested the **"hybrid"** trust approach for security implementations in the trust based systems.

With lots of people working on the trust based security, security systems were not confined to the traditional security systems only. But rather developments in Trust based frameworks [50] were serious contenders in which the automation of the administrating access control in web services started coming to light. The concept of honesty and dishonesty was by and large introduced. Even trustworthy computing was introduced by Microsoft after the serious problem in the Windows XP launch. The flaws were seriously taken by the hackers and exploited to a great extent.

A Global report **[51]** in the SOA/Web Services Security initiatives highlighted the security issues in the security of SOA/Web Services Security. Companies started investing heavily in Identify and Access Management (IAM). Also the major contributions started coming from the different researchers from all part of the world that with the adoption of Web Services, developers/administrators must cope up with the complexity of evolving trust negotiation policies spanning numerous autonomous services. The **Trust-Serv [52]** framework used a state machine based modeling approach that supports life cycle policy management and automated enforcement.

Research initiatives during the year 2010

A Robust Web Services in the Heterogeneous Military Networks suggested by Ketil Lund **[53]** and to be implemented in the NATO came into existence in 2010. It was suggested that the military operations should be based on the Service Oriented Architecture (Web Services). With the help of this we can ensure safe communication between the participating forces in the NATO. Because multiple countries are involved and they are going to access the information over the heterogeneous networks, the security was a major concern. So a system was developed which would generate the encrypted message of the actual SOAP Object and help in transmitting over the network.

Web services were another crucial factor in the development of a system in **Afyon Kocatepe University [54].** The System was called as WBDES and was compatible with any kind of Educational institution. It was based on the basic principle of Service Oriented Architecture and it utilized the authentication and authorization control in that. Here multiple users would login in the system and the web services at the backend were able to grant access rights and privileges to the users. The file transfer operation, user authentication, user authorization was purely done by the web services. But it lacked the trust factors which were not considered in the development of the system. And an administrator would be doing all the stuff manually, which over the period of time will look not feasible and involve lots of complexities.

A major research was proposed by Wanita Sherchan **[55]** which dealt with the security (authentication and authorization) on the trust basis. The trust service ontology which was proposed was purely on the basis of the reputation – based trust in the service web. The **ontology also uses the concept of community. The proposed system also defined the types of Trust like: Bootstrapped Trust, Global Trust, Personalized Trust, Direct Trust**, **Composite Trust etc**. In order to understand the system lets take an example. Consider a travel scenario of airline. Trust for the airline Domestic Jet Airways will be evaluated within the airline community. A service or a provider may be registered with more than one community. For example, the Jet airways may be registered to both domestic and international airline community. It would maintain separate trust levels in the separate communities. The system also proposed the bootstrapping trust i.e. the initial trust level so that the new node or entrant does not suffers because of his/her new entry. Rather a minimum number is suggested for the initial level of the trust. The framework

will also contain a Trust Manager that will receive all the requests for the authorization and trust manager gets the calculated trust level of the client and authorizes him/her accordingly.

Improving on the above mentioned framework, E.S. Shamila, Dr. V. Ramachandran **[56]** suggested another framework. They were in fact the first one who detailed the concepts of belief and trust. They suggested a framework that can easily grant and deny the requests **based on the direct, active, passive trust**. **The results were quite positive for the framework over a period of time the user interacts with the system**. There were also ingredients of the recommendations in their proposed methodology. A new layer of agreement was added by them. The agreement framework basically worked on the fundamental of providing the agreement regarding the web service to be developed. It means the web service when registered will also be registering the agreement with it. By this way, client can ask for the web service to be called with the necessary agreement to be invoked upon. The agreement works on trust and thus controlling authorization and authentication.

So it quite evident that a variety of frameworks were suggested which were based on the trust factor. All have helped in proposing frameworks that can help in providing a solution to be implemented in security of Web Services in Service Oriented Architecture. The problem of authentication and authorization has been addressed largely in every framework. But no serious work has been found in building of a framework that can maximize the authorization control based on trust, confidentiality, feedback and other factors in a typical web services environment.

3. OBJECTIVE AND SCOPE OF THE STUDY

3.1 Problem formulation (Motivation)

Ongoing research in the field of web services security (authentication and authorization) has led to the development of various frameworks and techniques. Researchers have suggested multiple methods of implementing security in web services. But there is a lack of development of systems that can automate security mechanisms in the area of web services. The researchers do talk about certain frameworks for authorization control but no serious framework or tool has been developed to ensure the web services access

control **[57]**. From the literature review it has been observed that most of the researchers are suggesting frameworks and designs for implementing security in web services through:

a. Token based.
b. Encryption
c. Web Services Trust Language

But very less amount of work has been carried out in automating the authorization control in web services. With the recent advancements in the SOA, there is a need for the system that can automate the process of **access control** and **policy based access** to these web services functions.

3.2 Gaps and Issues

As per the recent research in the field of security in the web services, still there is no concrete method of automating the process of authorization control in web services. These are serious challenges which every business organization faces in terms of administering the roles based access **[58]** to the web services. Because web services are publically available and they are easily discoverable with the help of WSDL, the access to these web services has to be restricted and managed by some administrator. With the exorbitant growth in the number of web services and its high adoption in the industry, the management of authorization control will become virtually impossible in the coming days. An administrator will have lots of limitations in terms of:

- ✓ Physical Limitations
- ✓ Limitations bases on time
- ✓ Geographical Limitations.
- ✓ Number of users.
- ✓ Type of accesses and restrictions.

Thus it clearly shows, that there is wide gap between the adoption of the web services as a standard way of writing the business applications and its access control which is done manually right now

3.3 Proposed Objectives

After a rigorous view of the literature and finding certain gaps/issues in the same, our proposed objective is to develop a framework that can maximize the automation of access control within web services. Our framework will replicate the idea of **"Trust" [55, 56] with a layer of "Confidentiality and Content",** which the human beings utilize to perform various actions. The idea is to create a framework that works automatically with some ingredients of intelligence to decide the authorization of any user. Creating such kind of system is not an easy task because the system regulated by the trust based mechanisms, has their independent set of problems. In long run, the trust based system may saturate to a situation where the trust basis becomes unstable. This would result in a complete distrust or the opposite i.e. blind trust. Our framework will try to find out a solution in between the complete distrust and blind trust. Neither of the two situations is suitable for the any business house. We need a method that helps in solving the overheads of the administration of access controls by using trust relationships and evaluate the capabilities of trust based access control. [59, 60]. Different types of frameworks [61, 62] have already been worked by different researchers but lack the potential of trust worthiness.

The following are prime objectives/scope of the study:

- **To develop a framework (collection of algorithms and federation of web services) that will automate the process of providing access control and restrictive access to the resources based on trust, confidentiality and content level.**
- **The framework will be based on various rules and regulations conditions that will govern the trust-level of the user.**
- **The framework will try to use the direct and recommended trust to achieve its objectives.**
- **The framework will try to minimize the problems with respect to the administration of access control of web services.**
- **To develop a generalized frame work that fits to any business domain.**

4. PROPOSED METHODOLOGY

4.1 General Research Methodology

In order to achieve the proposed objectives of writing a framework based on trust, Conclusive research design methodology will be followed. This approach will be based on formal and structured methodologies like:

a. *Specific hypothesis formulation based on review*.
b. Sample data collection and Experimentation
c. Q*uantitative Data Analysis*.
d. Results and Conclusion: Findings to be used an input into decision making

4.2 Experimentation

There are good amount of technologies available in the market like .NET/JAVA/IBM Rational Rose that can help in the writing the framework (algorithms) for maximizing the access the control in web services. These technologies support all the standards of Web Services, WS-Trust, WS-Federation Languages and they can help in developing interoperable solutions.

Experimentation will include the following tasks:

a. Framework will be developed with the help of standard tool sets like .NET/JAVA/Rational Rose etc. It will be a platform independent framework and can be deployed in any business house.
b. The framework will be totally based on web services.
c. The framework will be based on the major factor that human beings use in day to day life i.e. "Trust" + "Confidentiality and Content Level"
d. The framework will provide the flexibility to adjust the authorization at different levels. Thus making the authorization a blend of manual plus automated process.
e. **The framework will help in giving benefit and reducing the trust level on the basis of various rules like:**
    - Day to day interactions with system.
    - Experience in the organization.
    - Recommendations from other agents participating in the framework.
    - Types of documents and information with respect to the confidentiality and content level.

f. **The framework will include some punishment and reward for every transaction that results in non-granting and granting of access for any function in web service.**
g. **The framework will be deployed in any existing system like University Management System used in Lovely Professional University, Phagwara. The initial value of trust + confidentiality will be setup and based on certain parameters decided by competent authority.**
h. **The framework will employ a Trust Server that will be playing a crucial role in security i.e. Authentication and Authorization. The framework will be deployed on an array of Trust Servers (typically in a Network Load Balancing environment) so that when the no. of users increase, the framework does not crash and is able to handle the load of concurrent users.**
i. **The framework will contain multiple algorithms that will automatically calculate and recalculate the trust indices at the end of a defined period on the basis of already defined set of rules and regulations.**
j. **The framework will also include some boot-strapping [63] values, also known as start values which are given by a competent authority based on the designations and other parameters.**
k. **The framework will be run for a period of 6 months to 1 year span.**
l. **Also a separate data will be recorded for the same number of users using the web services without the framework.**

4.3 Results and Conclusion

a. At the end of the specific period of running the framework, the results of the Trust indices will be compared with the initial Trust indices.
b. The same Trust Indices will also be compared with the recording of data for the same users who have used the system without framework.
c. Because there are multiple parameters that are helping in effecting the Trust Indices (positively and negatively), system will automatically improve authorization control in web services for every user without intervention of any human effort.

d. The trust index results which the framework will provide can be evaluated against the HR appraisal results for the same period for any employee. Any user, whose trust index has improved over a period of time, will normally have a better HR appraisal for that period.

5. REFERENCES

[1] Alfred C.Weaver, "Enforcing Distributed Data Security via Web Services", IEEE International Workshop on Factory Communication Systems 22-24, Sep, 2004 pp. 397-402.

[2] Jinpeng Wei, Lenin Singaravelu and Calton, "A Secure Information Flow Architecture for Web Services Platforms", IEEE Transactions on Services Computing, Vol. 1, No. 2, April – June 2008.

[3] Laurel Reitman, James Ward, Jack Wilber, "Service Oriented Architecture (SOA) And Specialized Messaging Patterns", A technical White Paper published by Adobe Corporation USA, 2007

[4] Erin Cavanaugh, "Web services: Benefits, challenges, and a unique, visual development solution", A white paper published by Altova, Inc. USA, 2006.

[5] Z.A. Baig, K.Salah, "Multi-Agent pattern recognition mechanism for detecting distributed denial of service attacks", IET Information Security, Vol. 4, Issue 4, p.p. 333-343, 2010.

[6] S. Bertram, M. Boniface, M. Surridge, N.Briscombe, M. Hall-May, "On Demand Dynamic Security for Risk-Based Secure Collaboration in Clouds", 3rd IEEE International Conference on Cloud Computing, 2010.

[7] Z. Song, S. Lee and R. Masuoka, "Trusted Web Service", Pro. Second Workshop Advances in Trusted Computing, 2006.

[8] Pakpoom Prechapanich, Prasong Praneetpolgrang, Nutsapun Paopun, "The Development of Web Services Security Framework for Ministry of Information and Communication Technology", Proceedings of the Fourth International Conference on eBusiness, November, 19-20, 2005 Bangkok, Thailand.

[9] Ashish Kundu, Elisa Bertino, "A New Model for Secure Dissemination of XML Content", IEEE Transactions on Systems, MAN and Cybernetics – Part C: Applications and Reviews, Vol. 38, No. 3, May, 2008

[10] Gunnar Peterson, "Service Oriented Security Indications for Use", Journal of Computing in Science and Engineering, Vol. 7, Iss. 2, March, 2009.

[11] Geer, D. "Taking Steps to Secure Web Services", Journal of Computers, Vol. 36, Iss. 10, Oct, 2003.

[12] Youxiang Duanm Yontang Bao, Lijiang Pan, "A Secure Web Service Model Based on Combination of SOAP Registration Info and Token Proxy",

International Symposium on Computer Science and Computational Technology, 2008.

[13] Nan Guo, Tianhan Gao, Bin Zhang, "A Trusted Web Services Assessment Model Based on Six Dimensional QoS Framework and End to End Monitoring", IEEE Transactions on Web Services Computing, 2008.

[14] Anderson A. "Web Services Policies", Journal of Security and Privacy, Vol. 4. Issue 3., pp: 84-87, June, 2006.

[15] A Joint Security Whitepaper from IBM and Microsoft, "Security in Web Services World: A Propose Architecture and Roadmap", April, 2002, Version 1.0

[16] Karthikeyan Bhargavan, Cedric Fournet, Andrew D. Gordon and Riccardo Pucella, "TulaFale: A Security Tool for Web Services", International Symposium on Formal Methods for Components and Objects, Springer Verlag Berlin, 2004.

[17] Chen Li and Claus Pahl. (2003), "Security in the Web Services Framework", Proceeding of the 1st International Symposium on Information and Communication Technology, p.481-486

[18] Shiping Chen, John Zic, Kezhe Tang, David Levy, "Performance Evaluation and Modeling of Web Services Security", IEEE International Conference on Web Services ICWS, 2007.

[19] Dhafer Thabet, Lamia Hassine, Henda Ben Ghezala, "Situational Secure Web Services Design Methods", International Conference on Software Engineering Advances, 2007

[20] Dhafer Thabet, Lamia Hassine, Henda Ben Ghezella, "Toward Situational Secure Web Services Design Methods", IEEE Transaction on Service Computing, 2008.

[21] J Thelin, P. J. Murray, "A Public Web Services Security Framework Based on Current And Future Usage Scenarios", Proceedings of the International Conference on Internet Computing June 23-27, 2007 USA.

[22] Lalana Kagal, Massimo Paolucci, Naveen Srivinasan, Grit Denker, "Authorization and Privacy for Semantic Web Services", Journal of IEEE Intelligent Systems, Vol. 19, Issue. 4, July, 2004.

[23] Hristo Koshutanski, "An Access Control Framework For Business Process for Web Services", Proceedings of the 2003 ACM workshop on XML Security, p15-24

[24**]** Nan Guo, Tianhan Gao, Bin Zhang, "Trusted Assessment of Web Services Based on a Six Dimensional QoS Model", IEEE Asia-Pacific Services Computing, 2008.

[25] Zheping Wu and Alfred C. Weaver, "Using Web Service Enhancements to Bridget Trust Relationships", International Conference on Web Services, 2007.

[26] Federica Paci, Mourad Ouzzani, Massimo Mecella, "Verification of Access Control Requirements in Web Services Choreography", IEEE International Conference of Services Computing", 2008.

[27] Fahid Akhter, Maria Claudia Buzzi, Marina Buzzi, Barbara Leporini, "Conceptual Framework: How to Engineer Online Trust for Disabled Users", IEEE /WIC/ACM – International Conference on Web Intelligence and Intelligent Agent Technology – Workshops, 2009.

[28] R. Yahalom, B. Klein, "Trust Relationships in Secure Systems – A Distributed Authentication Prespective", Proceedings of the IEEE Symposium on Security and Privacy.

[29] Sharon Paradesi, Prashant Doshi and Sonu Swaika, "Integrating Behavioral Trust in Web Service Compositions", IEEE International Conference on Web Services, 2010.

[30] Center for Education and Research in Information Assurance and Security (CERIAS). *Formalizing Trust, Fraud, and Vulnerability, http://www.cs.purdue.edu/homes/bb/NSFtrust.html*, 2003.

[31] Hua Wang, Yanchun Zhang, Jinli Cao, "Achieving Secure and Flexible M-services through tickets", IEEE Transactions on System. MAN and Cybernetics Part A: Systems and Human, Vol. 33, No. 6 Nov. 2003.

[32] Ruchith Fernando, Dimuthu Leelarathne, Malinda Kaushalye, "SecureZone: A Framework for WS-Secure Conversation and WS-Trust", 2nd International Conference on E-Governance, 2004.

[33] Jia Zhang, Liang Jie Zhang,Jen Yao Chung, "WS-Trustworthy A Framework for Web Services Centered Trustworthy Computing", Proceedings of the 2004 IEEE International Conference on Services Computing.

[34] Zhengping Wu, Alfred C. Weaver, "Token-based Dynamic Trust Establishment for Web Services", Proceedings of the 43rd annual Southeast regional conference - Volume 2, ACM, New York, 2005.

[35] Satoshi Makino, Michiaki Tatsubori, Kent Tamura and Yuichi Nakamura, "Improving WS-Security Performance with a Template Based Approach", Proceedings of IEEE International Conference on Web Services", 2005

[36] Nils Agne Nordbotten, "XML and Web Services Security Standards", IEEE Communication Surveys and Tutorials Vol. 11, No. 3, Third Quarter, 2009.

[37] Zhengping Wu and Alfred C. Weaver, "Bridging Trust Relationships with Web Service Enhancements", IEEE International Conference on Web Services, p.p. 163-169, 18-22 Sep, 2006.

[38] Loubna Mehouar, Youssef Iraqi, "TrustWS- A Trust Management System for Web Services", IEEE Transactions on Services Computing 2007.

[39] Jia Zhang, "Trustworthy Web Services: Actions for Now", Journal of IT Professional, Vol. 7, Issue: 1, Jan – Feb, 2005.

[40] Ouzzani, M.; Bouguettaya, A, "Efficient Access to Web Services", Journal of Internet Computing (IEEE), Vol. 8, Iss: 2, pp: 33-34, April, 2004.

[41] Bhargav-Spantzel, A.; Squicciarini, A.C.; Bertino, E, "Trust Negotiation in Identity Management", Journal of Security and Privacy (IEEE) Vol:5, Iss: 2, pp. 55-63, April, 2007.

[42] Yan Wang Kwei-Jay Lin Macquarie, "Reputation – Oriented Trustworthy Computing in E-Commerce Environments", IEEE Journal of Internet Computing, Vol. 12, Iss: 4, pp: 55-59, July-Aug, 2008

[43] U. Wilhelm, S. Staamann, and L. Buttyan, "On the problem of trust in mobile agent systems", in Proc. IEEE Network Distributed Systems Security Sym. San Diego, CA, 1198, pp.11-13

[44] Walter Binder, "A Flexible Directory Query Language for the Efficient Processing of Service Composition Queries", International Journal of Web Services Research Vol. 4, p.p. 59-79, 2008.

[45] Marzouk.S.Mokbel, Le Jiajin, "Integrated Security Architecture for Web services and this Challenging", Journal Of Theoretical and Applied Information Technology", 2008 Issue.

[46] Nian Liu, Jianhua Zhang, "A Security Mechanism of Web Services-Based Communication for Wind Power Plants", IEEE Transactions on Power Delivery, Vol. 23, No. 4, October 2008.

[47] Jamal Bentahar, Babak Khosravifar, Maziar Gormokchi, "Social Network Based Trsut for Agent Based Services", IEEE International Conference on Advanced Information Networking and Applications Workshops, 2009.

[48] Alireza Zarghami, Soude Fazeli, Nima Dokoohaki, Mihhail Matskin, "Social Trust Aware Recommendations System: A T-Index Approach", IEEE/ACM/International Conference on Web Intelligence And Intelligence Agent Technology, 2009.

[49] Nicola Dragoni, "Toward Trustworthy Web Services Approaches, Weakness and Trust –By – Contract Framework", IEEE/ACM/International Conference on Web Intelligence and Intelligent Agent Technology, 2009.

[50] George Spanoudakis, Stephane LoPresti, "Web Service Trust: Towards A Dynamic Assessment Framework", IEEE International Conference on Availability, Reliability and Security, 2009.

[51] Gib Trub, Laurie Olski, "Global Report on SOA/Web Services Security initatives", GMG insights, September, 2008 Version 1.

[52] Halvard Skogsrud, Hamid R. Mothari-Nezhad and Boualem Bentallah, "Modeling Trust Negotiation For Web Services", Journal of Computers (IEEE) Vol. 42, Issue, 2, pp. 54-61, 2009.

[53] Ketil Lund, "Robust Web Services in Heterogeneous Military Networks", IEEE Communication Magazine, 2010.

[54] O. Deperlioglu, Y. Arslan, "Design principles of web-based distance education system and sample application in Afyon Kocatepe University", IET Software, Vol. 4, Iss. 4, pp. 283-293, 2010

[55] Wanita Sherchan, Surya Nepal, Jonahton Hunklinger, Athman Bouguettaua, "A Trust Ontology for Semantic Services", IEEE ICSC 2010.

[56] E.S. Shamila, Dr. V. Ramanchandran,"Evaluating Trust For Web Services Access", IEEE Transactions on Services Computing, 2010.

[57] I. Blanquer, V. Hernandez, D. Segrelles and E. Torres, "Enhancing Privacy and Authorization Control Scalability in the Grid Through Ontologies", IEEE Transactions on Information Technology in BioMedicine, Vol. 13, No. 1, January 2009.

[58] David W. Chadwick, Alexander Otenko, Edward Ball. "Implementing Roles Based Access Controls using X.509 Attribute Certificates", IEEE Internet Computing, March – April, 2003, pp. 62-69

[59] E. Damiani, S.D. Capitani, S. Paraboschi, and P. Samarati, "Design and implementation of an access control Processor for XML Documents", Computer and Networks Journal, Vol. 33, no. 1-6, pp59-75, June, 2000.

[60] "A Fine grained access control system for XML Documents", ACM Trans. Information System Security, vol. 5, No. 2 pp. 169-202, May, 2002

[61] Hridesh Rajan and Mahantesh Hosamani, "Tisa: Toward Trustworthy Services in a Service Oriented Architecture", IEEE Transactions on Services Computing, Vol. 1, No. 4, October-December, 2008.

[62] Ali Nasrat Haidar and Ali E. Abdallah, "Towards A Formal Framework for Developing Secure Web Services", 2nd International Conference on Automated Specification and Verification on Web Systems, 2006.

[63] Zaki Malik, "Reputation Bootstrapping for Trust Establishment among Web Services", IEEE Journal of Internet Computing, Vol. 13, Iss:1, Jan, 2009.